

IN THE SPECIFICATION

Please replace the paragraph beginning at page 1, line 5, with the following replacement paragraph:

The present application claims priority to the U.S. provisional patent application identified by Serial No. 60/158,777 filed on October 12, 1999, the disclosure of which is incorporated by reference herein. The present application is related to (i) PCT international patent application identified as PCT/US99/23008 (attorney docket no. YO998-392) filed on October 1, 1999; (ii) PCT international patent application identified as PCT/US99/22927 (attorney docket no. YO999-111) filed on October 1, 1999; (iii) PCT international patent application identified as PCT/US99/22925 (attorney docket no. YO999-113) filed on October 1, 1999, each of the above PCT international patent applications claiming priority to U.S. provisional patent application identified as U.S. Serial No. 60/102,957 filed on October 2, 1998 and U.S. provisional patent application identified as U.S. Serial No. 60/117,595 filed on January 27, 1999; and (iv) U.S. patent application identified as U.S. Serial No. 09/507,526 (~~attorney docket no. YO999-178~~) filed on February 18, 2000 which claims priority to U.S. provisional patent application identified as U.S. Serial No. 60/128,081 filed on April 7, 1999 and U.S. provisional patent application identified by Serial No. 60/158,777 filed on October 12, 1999. The disclosures of all of the above-referenced related applications are incorporated by reference herein.

Please replace the paragraph beginning at page 8, line 15, with the following replacement paragraph:

NL is a statement which is not limited to speech but encompasses all aspects of a natural multi-modal conversational application. It combines NL inputs with natural multi-modal input. As described in the above-referenced PCT international patent application ~~identified by attorney docket no. YO999-111~~ PCT/US99/22927 : any input is modeled independently of the modality as an input/output event that is then processed by a dialog manager and arbitrator that will use history, dialog context and other meta-

information (e.g., user preference, information about the device and application) to determine the target of the input event and/or engage a dialog with the user to complete, confirm, correct or disambiguate the intention of the user prior to executing the requested action.

Please replace the paragraph beginning at page 10, line 9, with the following replacement paragraph:

It is to be noted that the term “CML” is used in the above-referenced PCT international patent application[[s]] identified by attorney docket nos. YO998 392 and YO999-178 PCT/US99/23008 and U.S. patent application 09/507,526. In these applications, the term is meant to refer to a declarative way to describe conversational interfaces. In accordance with the present invention, the term CML refers to a gesture-based language which embodies the concept of programming by interaction, as will be explained in detail below.

Please replace the paragraph beginning at page 26, line 3, with the following replacement paragraph:

(ix) When the transcoding is performed by a multi-modal/conversational browser (as described below), the gestures are uniquely identified using a node_id tag. This allows not only to produce the rendering in each registered modality (local or distributed), but also to provide very tight synchronization (i.e., on a gesture level or even sub-gestures levels, when it is a gesture for which this makes sense). For example, an event (I/O event) immediately impacts the state of the dialogs (i.e., the state as maintained in the multi-modal shell, for example, as in the above-referenced U.S. patent application identified by attorney docket no. YO999-178 09/507,526) and the other modalities. Thus, such tight synchronization may exist between the HTML rendering 12 as may be supported by a personal digital assistant and the VoiceXML rendering 16 as may be supported by a conventional telephone.

Please replace the paragraph beginning at page 34, line 18, with the following replacement paragraph:

In addition, it is also possible to declare this processing through an object tag, e.g., <object> . . . <object>. An object tag allows for loading Conversational Foundation Classes (CFCs) or Conversational Application Platform (CAP) services (see, e.g., the above-referenced PCT international patent application identified as PCT/US99/22927 (attorney docket no. YO999-111 wherein CAP is equivalent to CVM or Conversational Virtual Machine). Arguments can be passed to the object using XML attributes and variables. Results can be returned via similar variable place-holders. This allows these objects calls to access and modify the environment.

Please replace the paragraph beginning at page 35, line 5, with the following replacement paragraph:

All the information needed to distribute the processing is described in the above-referenced PCT international patent application identified as PCT/US99/22925 (attorney docket no. YO999-113) which defines an architecture and protocols that allow distribution of the conversational applications. As such, the international patent application describes how such distribution can be done and how it allows, in the current case, to distribute the processing between a client browser and a server browser, as well as between local engines and server engines. This allows distribution of the processing of the input/output event across the network.

Please replace the paragraph beginning at page 40, line 12, with the following replacement paragraph:

Input events that are not handled by CML gestures making up the application bubble up to the CML interpreter where standard platform events such as help are handled by a default handler. Bubble up means that search of a gesture that matches the trigger value is hierarchically bubbling up from the closest enclosing gesture to a higher

one, until no gesture matches. In such a case, the trigger should be associated to a service offered by the browser, if not by the underlying platform (e.g., conversational virtual machine of YO999-114 PCT international patent application PCT/US99/22927). If none are met, the event is ignored or a default message is returned to the user explaining that the input was not understood (or not supported) and ignored. These, however, are implementation choices of the browser and underlying platform, not choices of the language. Note that mechanism bind-event is designed to override platform behavior -- it is not meant to be used as the exclusive mechanism for mapping user input to CML gestures. Thus, using element bind-event to bind all valid spoken utterances in an application to the appropriate gestures is deprecated.

Please replace the paragraph beginning at page 41, line 21, with the following replacement paragraph:

Note that to activate groups of gestures in parallel is the way to implement mixed initiative NL interfaces: each command/query supported at a given time is characterized by a form built out of gestures (i.e., a group of gestures is called a form). When an input/output event occurs, the dialog manager provided by the browser or underlying platform will guess what are the gestures in the different forms that are activated and they allow to qualify their associated attributes (the environment variables associated to the gestures). When all the mandatory attributes of a form have received a value, the action is considered as disambiguated and executed. Note that extra constraints between the attributes can be expressed using XFORMS, as will be explained below. See also the above referenced PCT international patent application identified by attorney docket no. YO998-392 PCT/US99/23008 for discussion on parallel activation, and K.A. Papineni et al., "Free-flow dialog management using forms," Proc. Eurospeech, 1999, and K. Davies et al., "The conversational telephony system for financial applications," Proc. Eurospeech, 1999, the disclosure of which is incorporated by reference herein.

Please replace the paragraph beginning at page 45, line 23, with the following replacement paragraph:

The process of transforming CML instances to modality-specific representations such as HTML may result in a single CML node mapping to a collection of nodes in the output representation. To help synchronize across these various representations, CML attribute node_id is applied to all output nodes resulting from a given CML node. When a given CML instance is mapped to different representations, e.g., HTML and VoiceXML by the appropriate modality-specific XSL rules, the shape of the tree in the output is likely to vary amongst the various modalities. However, attribute node_id allows us to synchronize amongst these representations by providing a conceptual backlink from each modality-specific representation to the originating CML node. In the above-referenced U.S. provisional patent application identified as U.S. Serial No. 60/128,081 (attorney docket no. YO999-178), a description is provided of how to develop a platform (the multi-modal shell) able to support tight multi-modal applications. The mechanism operates as follows. Each modality registers with the multi-modal shell the commands that it supports and the impact that their execution will have on the other registered modalities. Clearly, in the current case, upon parsing the CML page and transcoding the gestures, each gesture is kept in a data structure (i.e., the table) in the multi-modal shell. Upon an I/O event in a given modality, the node_id information is used to find the activated gesture and from the table (i.e., the CML document dialog tree), it is immediate to find the effect on the activated modality as well as the other modality (i.e., update of each view or fetch of a new page on the CML server).

Please replace the paragraph beginning at page 106, line 22, with the following replacement paragraph:

Before describing multi-modal browsing according to the present invention, the following is a summary description of some of the above-referenced patent applications with concepts relating to CML and the multi-modal browser of the present invention. For

~~ease of reference, the related applications are referred to via their respective attorney docket numbers.~~

Please replace the paragraph beginning at page 106, line 27, with the following replacement paragraph:

YO999-111 PCT international patent application PCT/US99/22927 discloses the concepts of: conversational computing, conversational user interface, and conversational application platform (CVM - Conversational Virtual Machine). The functionalities and behavior/services described therein YO999-111 and provided by CVM can be, in practice, implemented by the multi-modal browser of the invention, or by applications which offer a conversational user interface. However, at a conceptual level, it is assumed that CVM implements all the necessary services to support the browser of the invention.

Please replace the paragraph beginning at page 107, line 6, with the following replacement paragraph:

YO998-392 PCT international patent application PCT/US99/23008 discloses the use of a declarative programming language (referred to as “CML” but which is different than the language of the invention) to program a conversational application (i.e., multi-modal). The YO998-392 language disclosed therein is a declarative language that supports the multi-modal/conversational user interface. In practice, the example/embodiment provided therein consists of ML pages written according to the “multiple authoring” model instead of single authoring as provided for in accordance with the present invention. Different examples of the declarative programming language where taught:

Please replace the paragraph beginning at page 107, line 26, with the following replacement paragraph:

YO999-178 U.S. patent application 09/507,526 describes a generic multi-modal shell. It describes how to support and program synchronized multi-modal applications (that they be declarative, imperative or hybrid). It uses registration tables where each application modality registers its state, the commands that it supports and the impact of these commands on the other modality. Again, no teaching of gestures and single authoring. An embodiment describes the architecture when the application is a browser (i.e., a browser associated to the rendering of each modality) and the shell receives a CML page (as defined in YO998-392 international application PCT/US99/23008), builds the registration tables and therefore synchronizes across the modalities.

Please replace the paragraph beginning at page 117, line 1, with the following replacement paragraph:

FIG. 15 illustrates the different steps performed by a CML multi-modal browser according to one embodiment of the present invention. When a CML page is fetched by the browser, the browser parses the CML content, e.g., similar in operation to an XML parser (step 90). The browser builds an internal representation of the interaction (i.e., the graph/tree of the different gestures described in the page) and the node-id. Using the gesture XSL transformation (or other transformation mechanisms like Java Beans or Java Server Pages) stored in the browser (block 98), it builds (step 96) the different ML pages sent to each rendering browser (block 100). Upon I/O events in a modality, the effect is examined (step 92) at the level of the interaction graph (i.e., as stored in the MM shell Registration table (block 94) as described in YO999-178 U.S. patent application 09/507,526). Note that the gestures XSL transformation rules can be overwritten by the application developer indicating where they should be downloaded. They can also be overwritten by user, application or device preference from what would be otherwise the default behavior. New gestures can also be added, in which case, the associated XSL rules must be provided (e.g., a URL where to get them).

Please replace the paragraph beginning at page 119, line 19, with the following replacement paragraph:

(iii) Conversational Foundation Class: The conversational foundation classes where introduced in Y0999-114 PCT international patent application PCT/US99/22927 as being imperative dialog components that are independent of the modality and that can run in parallel and in series to build more complex dialogs. Combined with the services provided by the conversational application platform (CVM - conversational virtual machine), they allow programming of imperative conversational (multi-modal applications) by loading/linking to the libraries of these foundation classes that the platform provides. As each CVM platforms provides it, the application developer can use them and not worry about the rendering within the modality/modalities supported by the device and their synchronization. Accordingly, each gesture defined declaratively in the CML specification provided herein can have a imperative implementation (e.g., in Java) that can run in series (one after the other) or in parallel (more than one active – like more than one form active at a time). Programming in CFC is equivalent to programming imperatively by interaction: you use and link to the some imperative gesture, you hook it to the backend and connect the gesture together by conventional code. You may add some modality specific customization in this code or in the CFC arguments. Then, you let the platform (CVM or a browser that implements the same level of functionality) handle the rendering within the appropriate modality and appropriate synchronization between modality as hard coded in the foundation class. An example would be a case where all the foundation classes are provided as Java Classes. This allows extension of the programming by interaction model to Java applets or servlets, etc.